Software Complexity Measurement of Water Poverty Mapping Application with Function Point Method

Teguh Wahyono
Faculty of IT
Satya Wacana Christian University
Salatiga, Indonesia
teguh.wahyono@staff.uksw.edu

Benfano Soewito
Binus Graduate Program
Bina Nusantara University
Jakarta, Indonesia
bsoewito@binus.edu

Spits Warnars Harco Leslie Hendric
Doctor of Computer Science Program
Bina Nusantara University
Jakarta, Indonesia
shendric@binus.edu

Ford Lumban Gaol
Doctor of Computer Science Program
Bina Nusantara University
Jakarta, Indonesia
fgaol@binus.edu

***Abstract*—The emergence of various problems of the water crisis in Central Java, urge on need for developing the mapping application of the water poverty level in that area to help the Government and related parties in executing the decision making required. This application will map the poverty levels of the water from 8,576 villages, 573 subdistricts and 35 regencies in Central Java by using the technique of spatial data mining and the approach of the water poverty index (WPI). Seeing the magnitude of that application scope, then it needs to do the software complexity measurement to find out further the resources needed for the development of the system in the future. The measurement of the software complexity is held with the Function Point (FP) method that produces the estimation of the resource requirements such as project work effort, project duration and speed of delivery. The results of the research showed that the software water poverty mapping owns the complexity level with FP at 136.64, the project work effort at 1,273 hours, the project duration in 4.037 months and the speed of delivery at 33.85 FP per person month.**

Keywords— Function Point Analysis; Water poverty mapping; Software Complexity measurement.

I. INTRODUCTION

In the recent years, the Province of Central Java has been experiencing the problems of the shrinkage of public access for fresh water. The quality of the raw water undergoes the degeneration, due to the high activity of the industry and the households which generate waste and other pollution sources [1, 2]. As many as 35 Watersheds (DAS) in Central Java undergo the water crisis and 136 contaminated. This matter can be understood because some of the watersheds start in switching function into a residence or a factory that unequaled with the good waste management system. Tempo [3] noted that 10 areas in Central Java started to experience the water crisis. Some areas that previously were abundant with water begin to undergo the signs of the water source's reduction. Jocom [4] noted the phenomenon of the "loss of" a number of springs in town of Salatiga. The town which was in 2004 noted owning 64 water springs, in 2012 only owns 32 water springs. Meanwhile, Karanganyar Regency also experiences the similar thing. The Regency which is located at the foot of Mount Lawu for the first time, in 2014, experienced a declining in water debit [5]. 30 of the spring water points experienced a quite significant declining in water debit from 515 liters per second into 150 liters per second.

The various problems about the water above obviously will have an impact on the cases of the food crisis, life safety and society welfare, so that it needs immediately contemplated about solutions to overcome it. With that background, the Center of Computation Study of Satya Wacana Christian University working with the Ecological Agency of the Central Java Province will develop the application of the water poverty mapping system to impart a depiction of an estimate of the water availability in the province of Central Java. With the existence of those mapping, it is hoped the condition of water resources could be controlled so that its availability could meet the needs of the human population either at this time or in the future.

The application of the water poverty mapping will analyze 8,576 villages, 573 sub districts and 35 regencies/cities in Central Java. The characterization of the region is done by using the technique of spatial data mining that will process the data either spatial or non spatial which collected starting from the village level up to the provincial level. The spatial data meant is in the form of the map of the borders of a village of Central Java. While the non spatial data were processed from the data of the village potential, the statistics of regencies/ cities in numbers, and the profile of natural resources. The measurement of the poverty levels of the water uses the variables in the Water Poverty Index (WPI) [6]. This application can be equipped with technology application such as Data Warehouse technology [17,18] and data mining technology such as using Attribute Oriented Induction technique [19,20] or AOI-HEP data mining technique for finding frequent or similar patterns [21,22,23].

Seeing the scope magnitude of the water poverty mapping system above, then it needs to do the software complexity measurement to find out the resources needed in detail if we want to build the application. This paper will discuss the software complexity measurement using Function Point Analysis (FPA) method. FPA is a method developed to calculate and to estimate the size of the system based on the functionality of the system [7]. FPA can also generate the estimation of resource requirements such as project work effort, project duration and speed of delivery, so that it becomes a consideration for the activity of the application development management in more details [7, 8].

II. Function Points Method

A. Function Point Analysis (FPA)

Function Point Analysis (FPA) for software sizing developed in the late 1970s at IBM by Allan Albrecht [7, 8]. The International Function Point Users Group (IFPUG) was founded in 1980 and produced its own FPA Counting Practices Manual (CPM). In 2000, IFPUG produced Release 4.1 of its CPM [8].

FPA is a method of quantifying the size and complexity of a software system according to the functions that the system delivers to the user [7,9]. This method measures software size by quantifying the functionality based on logical design and functional specifications. This measure has great advantages: (1) independent from the programming language, development methodology, technology or capability of the project team used to develop the application, (2) comprehensible for client and user, and (3) applicable at an early phase of software development [7, 9].

B. Step by Step Calculation of Function Point Analysis

To measure the software complexity estimation using FPA approach following procedure as shown in Figure 1.

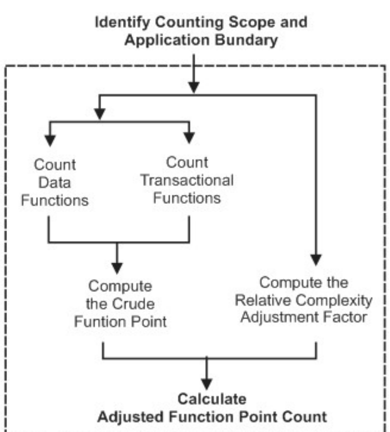

Fig. 1. FPA Procedure at a Glance [10]

1. Identify the counting scope and application boundary.

The first step in calculating FP is to identify the counting boundary. Counting boundary is the border between the application or project being measured and external applications or the user domain [10, 11]. Figure 2 shows FPA components. The system boundary establishes which functions are included in the function point count.

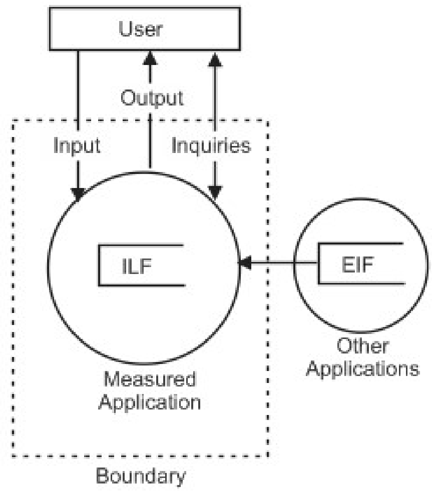

Fig. 2. FPA Components [10]

2. Compute the unadjusted function count or Crude Function Points (CFP).

   Crude Function Point (CFP) or unadjusted function count, relates to the following five software components: (1) number of user inputs, (2) number of user outputs, (3) number of user online queries, (4) number of logical files, and (5) number of external interfaces [7, 8, 10]. As shown in Table 1, weighted factors are applied to each component according to their complexity level.

TABLE I. CFP Calculation Form [7, 10]

Software System Component	Complexity Level[a]									CFP
	Simple			*Average*			*Complex*			
	C	*W*	*P*	*C*	*W*	*P*	*C*	*W*	*P*	
User Inputs		3			4			6		
User Outputs		4			5			7		
Queries		3			4			6		
Logical Files		7			10			15		
Ext. Interfaces		5			7			10		
Total CFP										

a. C = Count, W=Weight, P=Point (C*W)

3. Compute the Relative Complexity Adjustment Factor (RCAF) for the project.

RCAF summarizes the complexity characteristics of the software system and varies between 0 and 70. This phase assign grades (0 to 5) to the 14 subjects that substantially affect the required development efforts. RCAF formula is [7, 8, 10]:

$$\text{RCAF} = \sum_{i=1}^{14} Si \quad (1)$$

where S is subjects that substantially affect the required development efforts. Table 2 shows the RCAF form.

TABLE II. RCAF FORM [7, 10]

No	Subject	Grade
1	Requirement for reliable backup and recovery	0 1 2 3 4 5
2	Requirement for data communication	0 1 2 3 4 5
3	Extent of distributed processing	0 1 2 3 4 5
4	Performance requirements	0 1 2 3 4 5
5	Expected operational environment	0 1 2 3 4 5
6	Extent of online data entries	0 1 2 3 4 5
7	Extent of multi-screen or multi-operation online data input	0 1 2 3 4 5
8	Extent of online updating of master files	0 1 2 3 4 5
9	Extent of complex inputs, outputs, online queries and files	0 1 2 3 4 5
10	Extent of complex data processing	0 1 2 3 4 5
11	Extent that currently developed code can be designed for reuse	0 1 2 3 4 5
12	Extent of conversion and installation included in the design	0 1 2 3 4 5
13	Extent of multiple installations in an organization and variety of customer organizations	0 1 2 3 4 5
14	Extent of change and focus on ease of use	0 1 2 3 4 5
	Total = RCAF	

4. Compute the number of Function Points (FP):

   The RCAF determines the Technical Complexity Factor (TCF) [7, 10]. TCF formula is:

   $$\text{TCF} = 0.65 + 0.01 * \text{RCAF} \quad (2)$$

   The number of function point formula:

   $$\text{FP} = \text{CFP} \times \text{TCF} \quad (3)$$

   where CFP is crude function point count, and TCF is technical complexity factor.

III. WATER POVERTY MAPPING APPLICATION: A CASE STUDY

A. System Design

As had been discussed previously, the Water Poverty Mapping system is an application which intends on mapping the poverty level of the water in the territories in Central Java province. System design is shown in Data Flow Diagram as in Figure 3, where clearly visible the components of input, output and the processes that occurred within it.

From Figure 3 it can be seen that this application is processing input data from two sources, those are from the operator and other applications. Another application meant is a profile application of the region either sub districts, provinces or regencies that are owned by the Central Bureau of Statistics, whereas the output of the application is used by the governments as a basic of taking the important policies related preservation and water resources development.

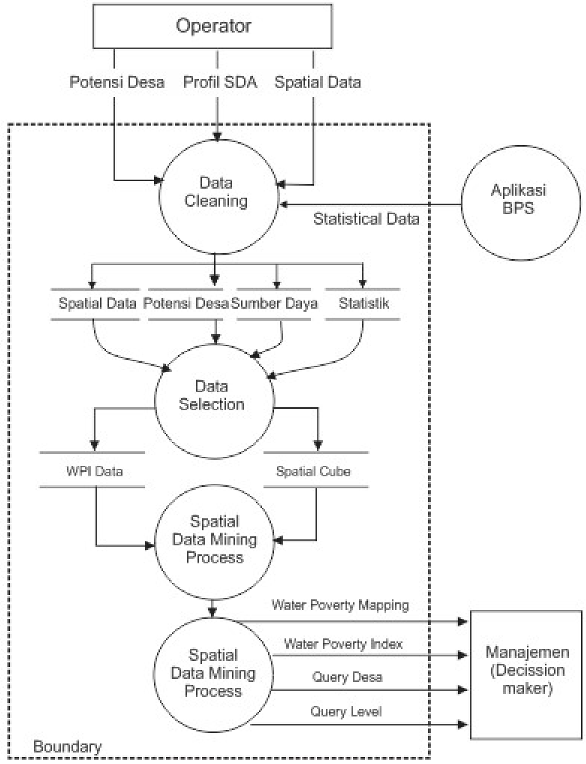

Fig. 3. Summary of Data Flow Diagram

B. Function Point Analysis

According to DFD is shown on Figure 3, it can be analyzed the quantities of the components on the application as follows:

- Number of user online queries : 2
- Number of logical files : 6
- Number of external interfaces : 1

Based on the degree of complexity of each of its components, then generated the FCP calculation as in table 3 below.

TABLE III. CFP CALCULATION

Software System Component	Complexity Level									CFP
	Simple			*Average*			*Complex*			
	C	*W*	*P*	*C*	*W*	*P*	*C*	*W*	*P*	
User Inputs	-	3	-	1	4	4	2	6	6	16
User Outputs	-	4	-	-	5	-	2	7	14	14
Queries	-	3	-	1	4	4	1	6	6	10
Logical Files	1	7	7	1	10	10	1	15	15	62
Ext. Interfaces	-	5	-	-	7	-	2	10	20	10
Total CFP										112

Afterwards, the evaluation toward the complexity characteristics of this application and the calculation of relative complexity adjustment factor described in table 4 below.

TABLE IV. RCAF CALCULATION

No	Affecting Subject	Grade
1	Requirement for reliable backup and recovery	4
2	Requirement for data communication	5
3	Extent of distributed processing	4
4	Performance requirements	3
5	Expected operational environment	4
6	Extent of online data entries	5
7	Extent of multi-screen or multi-operation online data input	5
8	Extent of online updating of master files	5
9	Extent of complex inputs, outputs, online queries and files	5
10	Extent of complex data processing	3
11	Extent that currently developed code can be designed for reuse	3
12	Extent of conversion and installation included in the design	3
13	Extent of multiple installations in an organization and variety of customer organizations	4
14	Extent of change and focus on ease of use	4
		57

Based on formula (1) and RCAF calculation on table 4, the number of RCAF is 57. The end calculation of the function point for the application of the Water Poverty Mapping based on formula (2) and (3) is as follows:

TCF = 0.65 + 0.01 * RCAF = 1.22

FP = 86 X 1.22 = 136.64

IV. RESULT AND ANALYSIS

The measurement of the software volume is an important issue for the developers to be able to plan the resources, the costs and the duration needed in building the software [12, 14]. Based on the results of the FP calculation which had been done, then it can be used to make an estimation of the resources required to undertake the development of the system. Following is the result of the estimated calculation of the resources required for the development of Water Poverty Mapping based on a formula developed ISBSG [11, 13, 16].

A. Project Work Effort (PWE)

PWE is a quantity that indicates the number of working time which is required for the development of the software project [11, 15]. The formula for PWE determination is as follows:

$$PWE = C \times Size^{E1} \quad (4)$$

Where PWE was a normalized Effort Project for the developers team (in units of Hours), the Size is the software size (FP), C is a constant (23.25) and E1 is a constant (0.814). From the formula (4), then the value of the PWE is as follows:

$$PWE = 23.25 \times 136.64^{0.814}$$

$$= 1{,}273 \text{ hours.}$$

B. Project Duration (PD)

The project duration shows the number of time needed for the completion of the software project [11, 16]. Project duration is calculated using the following formula:

$$PD = C \times Size^{E1} \quad (5)$$

Where the Duration is the active time of the working of the software project (months), Size is the software size (FP), C is a constant (0.543) and E1 is a constant (0.408). From the formula (5), then the value of the PWE is as follows:

$$PD = 0.543 \times 136.64^{0.408}$$

$$PD = 4.037 \text{ month.}$$

C. Speed of Delivery (SD)

Speed of delivery shows the speed of the working of the software project by the whole developer team [11, 16]. The formula to measure the speed of delivery is as follows.

$$SD = C \times Size^{E1} \quad (6)$$

Where SD is the speed of delivery (FP per person month), Size is the software size (FP), C is a constant (1.842) and E1 is a constant (0.592). From the formula (6), then acquired the value of SD as follows:

$$SD = 1.842 \times 136.64^{0.592}$$

$$SD = 33.85 \text{ FP per person month.}$$

By the calculations above, they can be used as the basic in executing the decision-making related to the further application development plan. Afterwards, the table 5 shows a summary of the water poverty mapping application measurement using Function Point Analysis.

TABLE V. SUMMARY OF WATER POVERTY MAPPING APPLICATION MEASUREMENT.

No	Description	Result
1	Crude Function Points (unadjusted function count)	112
2	Relative Complexity Adjustment Factor	57
3	Technical Complexity Factor	1.22
4	Function Point	136.64
5	Project Work Effort	1,273 hours
6	Project Duration	4.037 month
7	Speed of delivery	33.85 FP per person month

V. Conclusions

The measurement of the software volume is an important issue for the developers to be able to plan the resources, the costs and the duration needed in building the software. Function Point Analysis is a method used to obtain the complexity size of a software based on its functionality, that is the size that shows how big the function provided by the software to its users. The use of this method is capable to estimate the resource requirement, especially those related to project work effort, project duration and speed of delivery in the development of the software.

The case study about the measurement of the application complexity of Water Poverty Mapping Point generates the value of Function Point at 136.64. The value can be converted into a value that shows the estimated resource requirements in its development that is the project work effort at 1,273 hours, project duration in 4.037 months and speed of delivery at 33.85 FP per person month. For the future, the evaluation to the product quality after the software completed developed can be done by comparing the volume of the system with a large number of error (error-count) in the software processed. Meanwhile from a business perspective, the volume of software can be the basic for determining the value of the price of the software product involved.

Acknowledgment

The authors would like to thank the Center of Computation Study of Satya Wacana Christian University and the Ecological Agency of the Central Java Province for supporting this work and providing access to their data and generously answering many questions about their research.

References

[1] P. Saptono, "Kualitas air baku di Jawa Tengah merosot", in *Suara Karya Newspaper*, 18 January 2013.

[2] United Nations Development Programme (UNDP), "Human Development Report: Beyond Scarcity: Power, Poverty, and the Global Water Crisis". Palgrave Macmillan. New York, 2006.

[3] E. Faisol, "Sepuluh daerah di Jawa Tengah mulai krisis air", in *Tempo Magazine*, Ed. 07 Juli 2015.

[4] A. Kristijanto, H. Jocom, Asa tersisa dari sumber daya air yang terdera", Griya Media Publishing, Salatiga, 2013.

[5] Bramantyo, "Tiga puluh mata air di Karanganyar turun debit", in *Ekuatorial*, Ed. October 2014.

[6] Sullivan, C.A., J.R. Meigh et.al. "The Water Poverty Index: Development and application at the community scale." *Natural Resources Forum*, 2003.

[7] H. Colin, "The Function Point Method : A course material", Pearson Education Limited, Retrievable from www. cs. nott. ac. uk, 2014.

[8] IFPUG, "Function point counting practices manual," Release 4.1.1, 2000.

[9] M. A. Al-Hajri, Abdul Ghani, et al, "Modification of Standard Function Point Complexity Weights System". *Journal System Software* 74, 2 (January 2005), 195-206.

[10] A. HeydarNoori, "Function Point Analysis : Course Material", University of Waterloo Ontario, Canada. 2015.

[11] International Software Benchmarking Standards Group (ISBSG), "Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration", The McGraw-Hill Companies, Inc., 2011.

[12] H. K. Raju and Y. T. Krishnegowda, "Software Sizing and Productivity with Function Points," in *Lecture Notes on Software Engineering* vol. 1, No. 2, 2013.

[13] Neelam B. Singhal, and C. V. Srikrishna, "A Case Study to Assess the Validity of Function Points", in *World Academy of Science, Engineering and Technology* Vol:2 2008-06-27

[14] Karto Iskandar, Ford Lumban Gaol, Benfano Soewito, Harco Leslie Hendric Spits Warnars, "Software size measurement of knowledge management portal with use case point", *The international conference on Computer, Control, Informatics, and its Applications* (IC3INA 2016), 3-5 Oct 2016, Tangerang, Indonesia. http://situs.opi.lipi.go.id

[15] J. T. Joseph, "Role of Function Point as a Reuse Metric in a Software Asset Reuse Program", *in International Conference on Software Engineering Research and Practice (SERP)*, Las Vegas, Nevada, USA, 2011.

[16] Anie Rose Irawati, Khabib Mustofa, "Pengukuran Fungsionalitas Perangkat Lunak Menggunakan Metode Function Point Berdasarkan Dokumentasi Desain", *IJCCS*, Vol.7, No.2, July 2013.

[17] Harco Leslie Hendric Spits Warnars, Richard Randriatoamanana, "Datawarehouser: A Data Warehouse artist who have ability to understand data warehouse schema pictures", *IEEE TENCON 2016 (Technologies for Smart Nation)*, pp. 2207-2210, 22-25 Nov 2016, Singapore. http://site.tencon2016.focalevents.sg/

[18] Spits Warnars, "Perbandingan penggunaan Database OLTP (Online Transactional Processing) dan Data Warehouse", *Creative Communication and Innovative Technology (CCIT) journal*, Vol. 8 No. 1, pp. 83-100, September 2014. ISSN: 1978-8282.

[19] Spits Warnars, "Mining Patterns with Attribute Oriented Induction", *The International Conference on Database, Data Warehouse, Data Mining and Big Data (DDDMBD2015)*, Tangerang, Indonesia, pp. 11-21, 10-12 September 2015.

[20] Harco Leslie Hendric Spits Warnars, Muhamad Iskandar Wijaya, Hok Bun Tjung, Dendy Fransiskus Xaverius, Dedy Van Hauten, Sasmoko, "Easy understanding of Attribute Oriented Induction (AOI) characteristic rule algorithm", *International journal of Applied Engineering Research (IJAER)* 2016, vol. 11, No. 8, pp. 5369-5375. Source: http://www.ripublication.com/ijaer16/ijaerv11n8_03.pdf

[21] Harco Leslie Hendric Spits Warnars, "Using Attribute Oriented Induction High level Emerging Pattern (AOI-HEP) to mine frequent patterns", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, No.6, 2016.

[22] Spits Warnars, "Mining Frequent and Similar Patterns with Attribute Oriented Induction High Level Emerging Pattern (AOI-HEP) Data Mining Technique", International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), vol. 3, Issue 11, 2014, pp.266-276. EISSN : 2279-0047, ISSN: 2279-0055. Source : http://iasir.net/IJETCASpapers/IJETCAS15-247.pdf

[23] Spits Warnars (2014),"Mining Frequent Pattern with Attribute Oriented Induction High level Emerging Pattern (AOI-HEP)", *IEEE the 2nd International Conference on Information and Communication Technology (IEEE ICoICT 2014)*, Bandung, Indonesia, pp. 144-149, 28-30 May 2014.